



Universitat
Autònoma
de Barcelona



2445: Anàlisi i resolució de puzzles

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica realitzat
per

Xavier Sureda Rodrigo

i dirigit per

Gemma Sánchez Albaladejo

Bellaterra, 10 de Setembre de 2010

El sotasignat, **Gemma Sánchez Albaladejo**

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en

I per tal que consti firma la present.

Signat:

Bellaterra, 10 de Setembre de 2010

Index

1. Introducció	4
1.1. Presentació	4
1.2. Motivació Personal	4
1.3. Objectius	6
1.4. Planificació del Projecte	8
1.4.1. Fases de desenvolupament	8
1.4.1.1. Adquisició i segmentació	8
1.4.1.2. Extracció de característiques	8
1.4.1.3. Codificació i Comparació	9
1.4.1.4. <i>Matching</i>	10
1.4.2. Planificació Temporal i Costos	10
1.5. Estructura de la Memòria	13
2. Anàlisi de Requeriments	14
2.1. Adquisició de peces	14
2.1.1. Càmeres	14
2.1.2. Escàner	15
2.2. Segmentació de peces	16
2.2.1. Adquisició individual	16
2.2.2. Adquisició múltiple	16
2.3. Extracció de característiques	17

2.3.1.	Anàlisi per color	18
2.3.2.	Anàlisi per forma	19
2.3.2.1.	Estructura de Dades	20
2.4.	<i>Matching</i>	22
2.4.1.	Tècniques Satisfacció de Restriccions / AC-4	22
2.4.2.	Força Bruta	23
3.	Sistema de Resolució	24
3.1.	Adquisició i segmentació d'imatges	25
3.1.1.	Adquisició	25
3.1.2.	Segmentació	26
3.1.2.1.	Detecció de l'objecte	26
3.1.2.2.	Retall de l'objecte	28
3.2.	Extracció de Característiques	28
3.3.	Codificació i Comparació de la imatge	32
3.3.1.	Codificació	32
3.3.2.	Estructura de Dades	32
3.3.2.1.	Peça	33
3.3.2.2.	Mapa de Peces	34
3.3.2.3.	Mapa de costats	34
3.3.3.	Comparació	34
3.4.	<i>Matching</i>	37

3.5. Representació final	37
4. Resultats obtinguts	39
5. Conclusions i treball futur	44
6. Bibliografia	46

1. Introducció

1.1. Presentació

La visió per computador és un subcamp de l'Intel·ligència Artificial que té el propòsit de programar un computador per entendre una escena

Des que Ivan Edward Sutherland va presentar els Gràfics per Computador per primer cop l'any 1963, la tecnologia i la investigació han portat a la creació de sistemes CAD, imatges en tres dimensions, realitat virtual i/o visió per computador.

L'ús de l'anàlisi informatitzat d'imatges es pot veure avui en dia aplicat a molts àmbits. Des del sanitari a l'hora de detectar anomalies cardíques, per exemple, fins a la seguretat aeroportuària per a buscar objectes no admesos, passant pel camp comercial on recentment es presentava un emprovador virtual d'ulleres.

El que plantegem en aquest projecte és el reconeixement, identificació i posada en correspondència d'un objecte bidimensional mitjançant tècniques de gràfics i visió per computador. En concret, l'objectiu és la resolució d'un puzzle.

1.2. Motivació Personal

D'una banda, els trencaclosques i els jocs lògics i d'enginy sempre han captat la meva atenció i interès. El fet d'intentar assolir nous reptes cada cop més ambiciosos o amb una dificultat afegida em motiva a resoldre puzzles amb major quantitat de peces, amb formes estranyes, o fins i tot els anomenats "impossibles", la imatge global dels quals és monòtona, com mostra la figura

1.

Per altre costat, el món dels Gràfics per Computador i la Visió per Computador, així com totes les seves aplicacions en el món real, també van despertar la meua curiositat des d'un principi.

D'aquesta manera plantejar-me resoldre un puzzle amb tècniques de Visió per Computador, a part del repte i la dificultat que això comporta, significa la unió de dues de les meves passions.



Figura 1: Puzzle de la col·lecció “Impossibles” de Educa Borràs

1.3. Objectius

L'objectiu principal d'aquest projecte és la resolució automàtica d'un puzzle típic de peces tetragonals on cada un dels costats es pot classificar a partir de tenir un tros que sobresurt, la manca d'un tros, o per ser recte. Per aconseguir-ho partim d'una sèrie d'imatges escanejades, com la mostrada a la figura 2, que contenen les peces que integren el trencaclosques.



Figura 2: Imatge obtinguda d'escanejar les peces d'un puzzle

Tot i centrar els esforços en el problema concret de la resolució dels puzzles ja esmentats, la finalitat és crear un sistema extrapolable a qualsevol objecte en dues dimensions. Per exemple, la reconstrucció d'un paper com el de la figura 3 es podria plantejar com a treball futur d'aquest projecte.



Figura 3: Papir dels manuscrits de Qumram

1.4. Planificació del Projecte

1.4.1. Fases de desenvolupament

El projecte s'ha planificat en quatre grans fases, cada una de les quals té una sèrie de tasques a desenvolupar.

1.4.1.1. Adquisició i segmentació

La primera consisteix en l'adquisició de les peces amb qualsevol dispositiu de captura d'imatges, i el seu emmagatzemament en un o diversos fitxers per a poder-ne fer l'anàlisi i etiquetar o identificar cada una de les peces que composaran el puzzle. El resultat d'aquesta part es mostra a la figura 4.

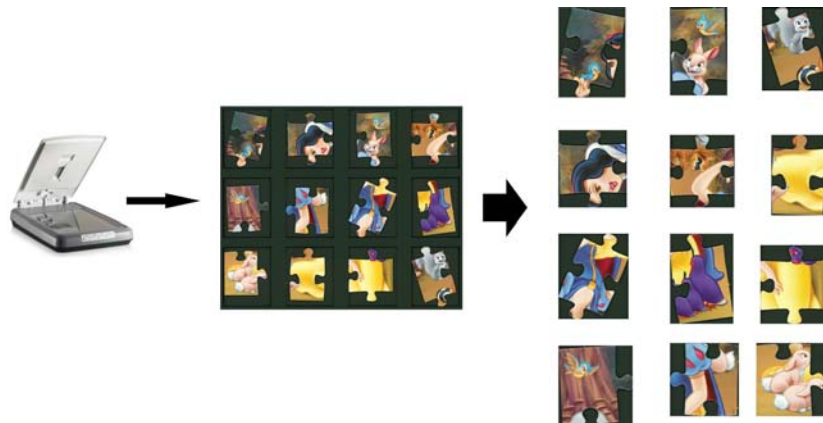


Figura 4: Resultat del procés d'adquisició i segmentació

1.4.1.2. Extracció de característiques

La segona fase del procés consisteix en el tractament de la imatge per tal d'eliminar el màxim possible de sorolls, ombres i altres alteracions que poden haver aparegut amb la captura. Seguidament se n'obté el contorn de cada

peça i se n'agafen les dades característiques. La figura 5 resumeix aquesta fase.

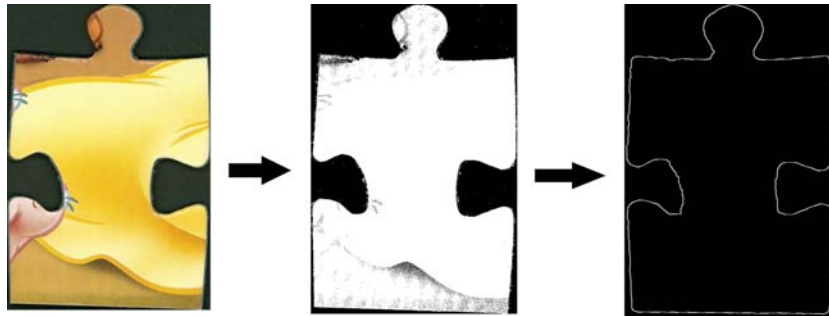


Figura 5: Resultat de la obtenció del contorn

1.4.1.3. Codificació i Comparació

La tercera part consisteix a codificar i construir una estructura de dades que defineixi i distingeixi cada peça de manera única, ja sigui amb dades de color o de forma, com mostra la figura 6.

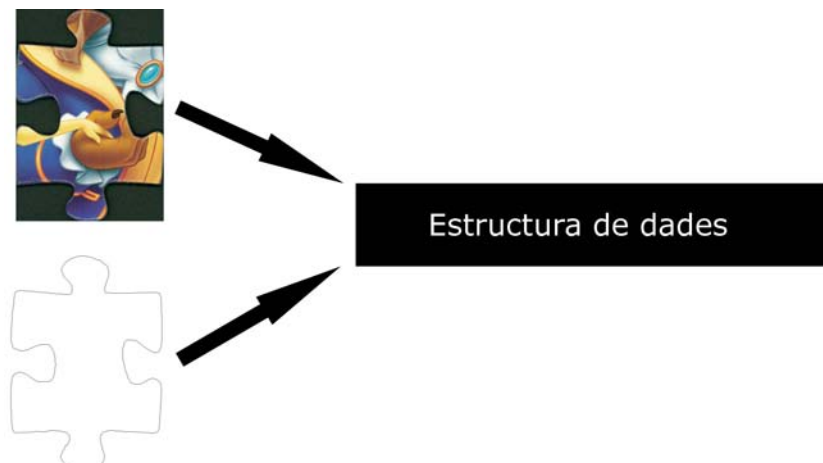


Figura 6: Construcció d'una estructura de dades a partir de les característiques de la peça

A continuació es comparen les diferents estructures per saber quin grau de similitud o de diferència té cada costat d'una peça amb tota la resta.

1.4.1.4. *Matching*

Aquest procés parteix dels valors obtinguts en la fase de la comparació i posa en correspondència les peces amb més provabilitat de ser connexes. Tal i com es proposa a [2], es comença per les quatre cantonades i el contorn, seguit de les peces interiors, partint d'aquelles de les quals ja en disposem dos costats contigus. El que se n'obté al final de tot es mostra a la imatge 7.



Figura 7: Resultat del procés de matching

1.4.2. Planificació Temporal i Costos

Les figures 8 i 9 mostren la planificació temporal i el cost en hores de cada una de les fases de desenvolupament, i el Diagrama de Gantt corresponent, dissenyat plantejant un horari laboral de 8h diàries i 5 dies a la setmana amb una sola persona com a recursos humans, respectivament.

Donat que tant el *hardware* utilitzat com les llicències del programari no són exclusius del projecte, com a cost només s'ha contemplat el salari d'un programador junior que és d'aproximadament 8'00 €/h. Això fa un total

de 7.120'00 €, sense tenir en compte l'alta de la seguretat social ni el que representa per una empresa tenir contractat a un treballador, o el pagament dels autònoms en cas de treballar per compte propi.

	Nom de la tasca	Cost	Inici	Duració
1	Projecte Final de Carrera	7.120,00 €	Mon 08/03/10	834 hours
2	Disseny	400,00 €	Mon 08/03/10	50 hours
3	Disseny i Anàlisi del projecte	400,00 €	Mon 08/03/10	50 hours
4	Recerca i lectura d'articles	192,00 €	Mon 08/03/10	24 hours
5	Proves i estudi de viabilitat	128,00 €	Thu 11/03/10	16 hours
6	Documentació prèvia	80,00 €	Mon 15/03/10	10 hours
7	Adquisició i Segmentació de peces	896,00 €	Tue 16/03/10	56 hours
8	Adquisició de Peces	64,00 €	Tue 16/03/10	8 hours
9	Escanejat de Peces	64,00 €	Tue 16/03/10	8 hours
10	Segmentació	384,00 €	Wed 17/03/10	48 hours
11	Detecció de les peces	128,00 €	Wed 17/03/10	16 hours
12	Retall de les Peces	64,00 €	Fri 19/03/10	8 hours
13	Obtenció del contorn	128,00 €	Mon 22/03/10	16 hours
14	Rotació de les Peces	64,00 €	Wed 24/03/10	8 hours
15	Extracció de característiques	384,00 €	Thu 25/03/10	48 hours
16	Extracció	384,00 €	Thu 25/03/10	48 hours
17	Detecció de 4 cantonades	64,00 €	Thu 25/03/10	8 hours
18	Dades característiques de forma	320,00 €	Fri 26/03/10	40 hours
19	Codificació i Comparació de dades	832,00 €	Fri 02/04/10	104 hours
20	Codificació de la peça	64,00 €	Fri 02/04/10	8 hours
21	Estructura de Dades	128,00 €	Mon 05/04/10	16 hours
22	Comparació	640,00 €	Wed 07/04/10	80 hours
23	Matching	2.368,00 €	Wed 21/04/10	296 hours
24	Procés de Matching	1.728,00 €	Wed 21/04/10	216 hours
25	Algoritmes de Matching	1.280,00 €	Wed 21/04/10	160 hours
26	Algoritmes de Poda	320,00 €	Wed 19/05/10	40 hours
27	Representació final	128,00 €	Wed 26/05/10	16 hours
28	Proves i depuració	640,00 €	Fri 28/05/10	80 hours
29	Test i avaluació del projecte	320,00 €	Fri 28/05/10	40 hours
30	Millores i canvis	320,00 €	Fri 04/06/10	40 hours
31	Finalització	2.240,00 €	Fri 11/06/10	280 hours
32	Presentació i documentació	640,00 €	Fri 11/06/10	80 hours
33	Documentació	960,00 €	Fri 25/06/10	120 hours
34	Preparació exposició	640,00 €	Fri 16/07/10	80 hours

Figura 8: Planificació i costos de les tasques del projecte

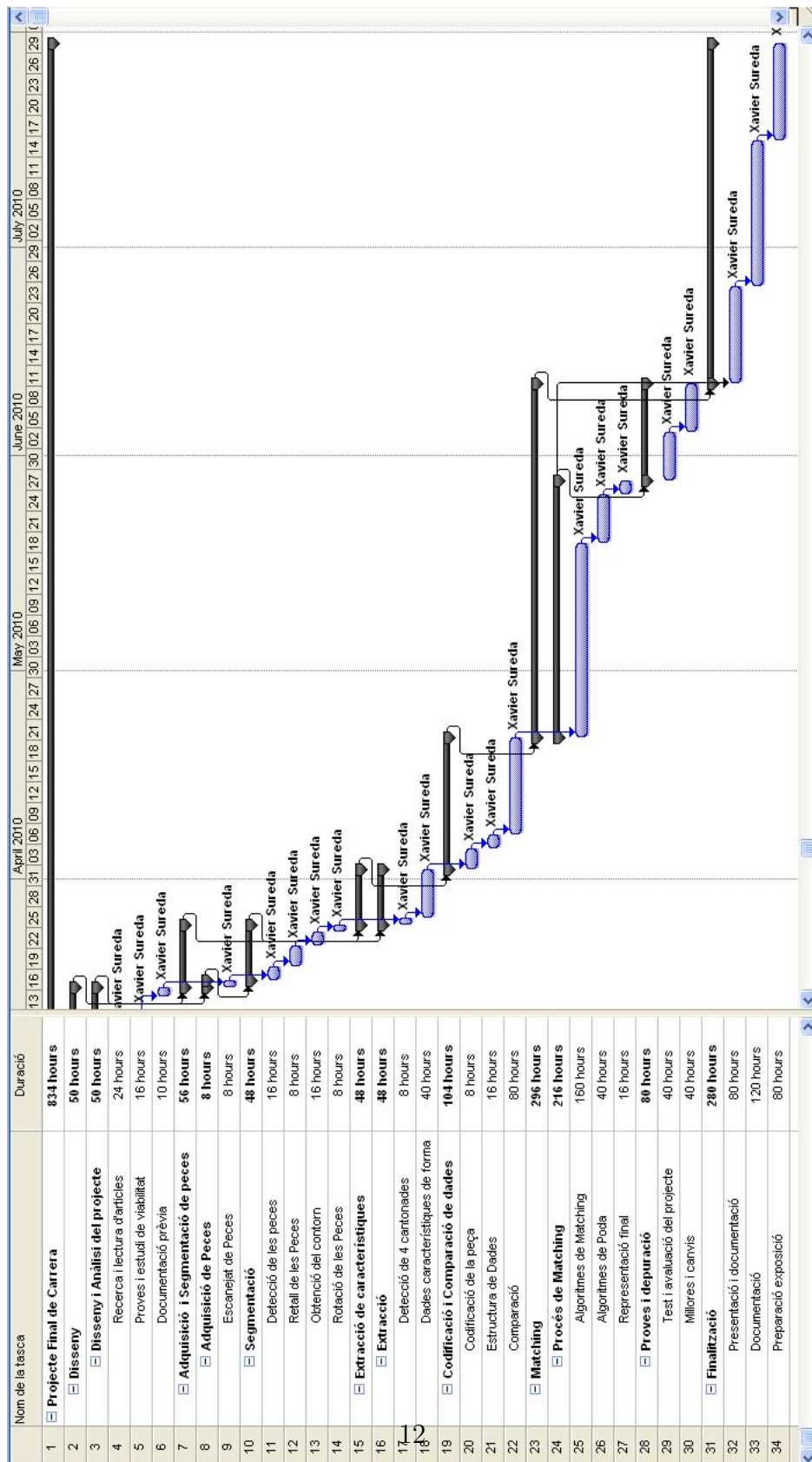


Figura 9: Diagrama de Gantt del projecte

1.5. Estructura de la Memòria

Aquesta primera secció de la memòria presenta el problema a resoldre i l'ordre que es segueix en el procés d'elaboració.

El segon apartat és un anàlisi de cadascún dels diferents aspectes que formen part del desenvolupament del programa, els recursos que s'han avaluat i els mètodes que s'han considerat previament i durant la realització del projecte final.

Seguidament, sota el títol de **3 Sistema de Resolució**, es descriuen amb detall els passos que s'han seguit i com s'han implementat, així com els resultats que es van obtenint pas a pas.

A **4 Resultats Obtinguts** es resumeixen les proves realitzades i la resposta en temps i fiabilitat obtingudes així com la seva valoració.

Finalment, dintre de **5 Conclusions i treball futur**, hi trobem una síntesi del treball realitzat, els possibles aspectes a reconsiderar i el rumb que pot prendre l'estat actual del projecte per a millorar.

Les última pàgina és per a la bibliografia utilitzada per al desenvolupament del projecte.

2. Anàlisi de Requeriments

El projecte s'ha dividit en una sèrie de fases. Cada una d'aquestes fases, a la vegada consta d'una sèrie de tasques. Donat que hi ha diferents mètodes, algorismes o sistemes per a complir les tasques, el primer que s'ha fet és un anàlisi de requeriments per a poder prendre la decisió de quin d'ells seguir o implementar. Els aspectes més crítics són els que es descriuen a continuació.

2.1. Adquisició de peces

La finalitat d'aquesta part és aconseguir una imatge amb una o varies peces. Els dispositius d'adquisició que s'han considerat són les càmeres fotogràfiques i els escaners, en gran mesura per ser fàcilment assequibles.

2.1.1. Càmeres

L'avantatge més important de l'ús de càmeres és la rapidesa a l'hora de capturar les imatges. Però els inconvenients recauen sobretot en crear un entorn controlat. Qualsevol canvi d'il·luminació pot causar ombres que alterin el resultat, la modificació de l'angle d'adquisició entre captures genera perspectives que augmenten considerablement la complexitat del problema o el soroll que pot ocasionar una vibració o una brutícia, com per exemple la pols, repercuteixen en els processos posteriors.

Tenint en compte aquest factors, aquesta opció es descarta per no poder garantir que les diferents imatges de les peces d'un mateix puzzle no tinguin cap dels problemes exposats.

2.1.2. Escàner

Utilitzant un escàner a l'hora d'obtenir les imatges, tenim l'avantatge que l'entorn controlat ja està creat i, si no hi ha un mal ús humà com el d'obrir la tapa de l'escàner durant el procés d'escaneig, la lluminositat és constant i igual per a totes les imatges, així com les ombres.

Els inconvenients amb els que ens trobem són, per un costat, el soroll causat per la superfície del escàner, ja sigui degut a la pols, ratllades i demés, i per altra banda les ombres que sempre apareixen amb els objectes que tenen gruix. Un exemple de brutícia i d'ombres generades es mostra a la figura 10.

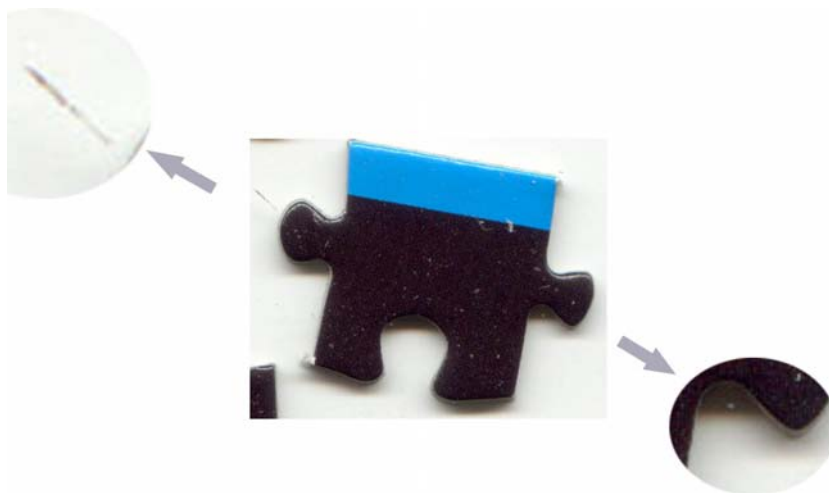


Figura 10: Soroll aparegut en el procés d'escanejat

Tot i els inconvenients, s'opta per aquest dispositiu, donat que és fàcilment assequible a tots els usuaris i a la senzillesa per a contrarestar aquests aspectes negatius. El punt més crític són les ombres, ja que mantinguent el dispositiu net i en bon estat, el soroll es pot reduir considerablement. Per tal d'evitar les ombres, el que hem de fer és posar un fons negre, ja sigui amb un cartró o cartolina o qualsevol altre material que no deixi passar la llum.

2.2. Segmentació de peces

L'objectiu és tenir una imatge per peça per tal que siguin fàcilment analitzables i tractables. Això ens deixa dues opcions, l'adquisició individual per cada peça o múltiples a la vegada.

2.2.1. Adquisició individual

L'opció d'adquirir cada peça individualment té el clar avantatge que tal i com obtenim la imatge del dispositiu ja es pot enviar a la següent fase del projecte. Així mateix, també té un gran inconvenient, el cost en temps o en recursos per a escanejar tantes vegades com peces tingui el puzzle. A més, una conseqüència que se'n deriva és l'augment de la provabilitat d'error humà, com ara moure la peça durant l'adquisició.

2.2.2. Adquisició múltiple

En el cas de l'adquisició de múltiples peces simultàniament, tenim com avantatge la reducció de temps per a escanejar totes les peces, i menys provabilitat d'error durant el procés, però l'inconvenient que un cop es disposa de la imatge, s'ha de dividir en múltiples sub-imatges, una per a cada peça.

Per tal de fer possible una bona detecció i retall de les peces, s'ha de tenir en compte l'espai de separació entre elles sobre la superfície de l'escàner. Dues peces cavalcades provoquen error, tant d'ombres, pel fet d'ocupar una alçada superior, com de detecció, ja que no es pot distingir quina part correspon a cada peça. Dues peces molt pròximes poden acabar considerant-se una de sola si les ombres o el color del contorn en els punts més pròxims s'assemblen. Per tant, si bé la orientació no té especial rellevància, la distància entre peces

és important.

La imatge 11 és part del resultat del procés de segmentació on no s'han tingut en compte les ombres que generen les peces i l'espai entre elles, provocant que es consideri el conjunt de les dues com una sola peça.



Figura 11: Dues peces detectades com una en el procés de segmentació

2.3. Extracció de característiques

El resultat d'aquesta part és la obtenció d'una sèrie de valors que han de configurar l'estructura de dades que defineix la peça de manera única i que ens ha de servir per a fer la comparació entre elles. La primera elecció que s'ha de fer és el tipus d'anàlisi de cada peça, que pot ser per color o per forma.

2.3.1. Anàlisi per color

Hi ha diversos algorismes basats en el color per a obtenir les característiques que poden adequiar-se a les necessitats del projecte. Però també trobem que gran part d'ells requereixen un anàlisi previ de la forma, de manera que una part d'algorísmica ha d'anar destinada a la forma. Així mateix, l'anàlisi per color és més ràpid pels casos on no hi ha grans extensions del mateix color, com un cel sense núvols o una gran extensió de gespa. D'altra banda, la complexitat augmenta i la posada en correspondència es complica en els casos de múltiples peces visualment iguals.

Un exemple d'aquest cas el trobem en la col·lecció de puzzles de les formigues de Fabio Vettori, on se'n mostra un detall a la imatge 12.

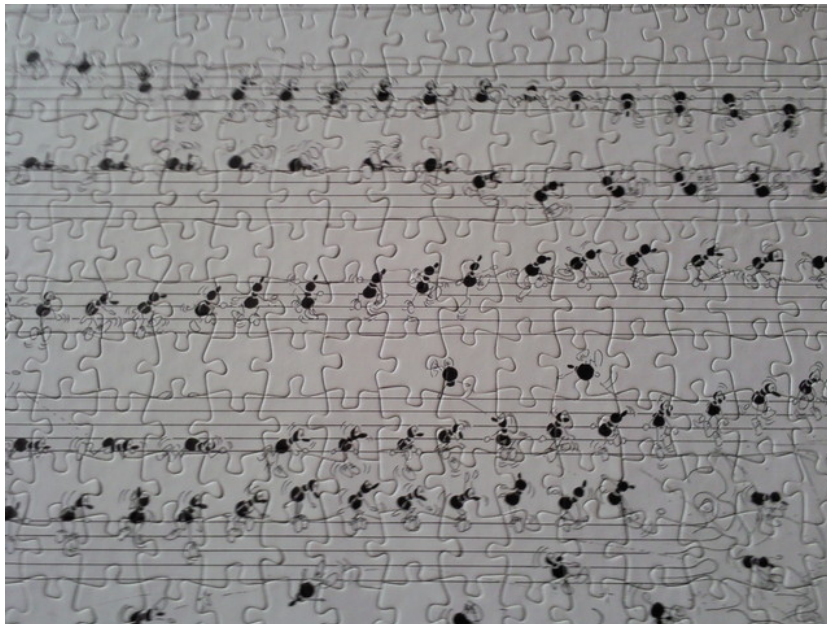


Figura 12: Detall d'un dels puzzles de la col·lecció de les formigues de Fabio Vettori

Basat en l'exposat fins aquí, els esforços en el desenvolupament i resolució del

projecte es destinen a un anàlisi per forma, deixant de banda l'anàlisi per color, considerant-lo un valor afegit i un recurs per a descartar possibles falsos positius, però que no serà tractat en aquest projecte, ja que en sí sol, aquesta metodologia es mereix un projecte sencer.

2.3.2. Anàlisi per forma

Així com passa pel cas de l'anàlisi del color, hi ha diversos mètodes i algorismes per analitzar la peça i obtenir una sèrie de característiques o trets distintius que poden posar de manifest la correspondència entre elles en funció de la forma. Generalitzant, es tracta de discriminar entre les quatre cantonades, les peces del contorn i les peces interiors i parametritzar de manera que siguin comparables, cadascún dels costats de la peça.

La problemàtica apareix quan els fabricants utilitzen el mateix patró per a tallar diferents peces de tot el puzzle. Fins i tot n'hi ha que tenen el patró simètric, és a dir, que la meitat de les peces són mirall de l'altra meitat. Per això, un anàlisi exclusivament per forma no sempre pot ser determinant o decisiu per aconseguir el resultat final. Un altre problema evident són aquells puzzles de mosaic els quals totes les peces són quadrangulars, de manera que no es pot fer la comparació per forma, com es veu a la figura 13.

Donat que com a precondició del projecte, s'ha determinat que el puzzle serà dels típics amb quatre costats que poden tenir un tros sortint (*outdent*) o un tros entrant (*indent*), s'opta per aquesta opció per al seu desenvolupament.

2.3.2.1. Estructura de Dades

Un cop escollit que l'anàlisi de la peça serà en base a la seva forma, s'ha de definir quines dades han de formar part de l'estructura de dades per a la posterior comparació. Les opcions a considerar han estat dues: la parametrització de la forma dels *Indents* i *Outdents* i la codificació dels costats en cadenes de caràcters.

2.3.2.1.1. Parametrització de la forma

L'avantatge dels algorismes que parametritzen la peça és que la quantitat de dades a emmagatzemar és mínima. Una de les maneres de definir la forma de l'*Indent* o l'*Outdent* és buscant el punt del centre de l'el·lipse inscrit a aquest, els punts tangents i els d'inflexió que el defineixen. La figura 14



Figura 13: Puzzle on les peces són cubs, tals que cada cara forma part d'una il·lustració diferent

mostra un exemple d'aquests punts en una peça.

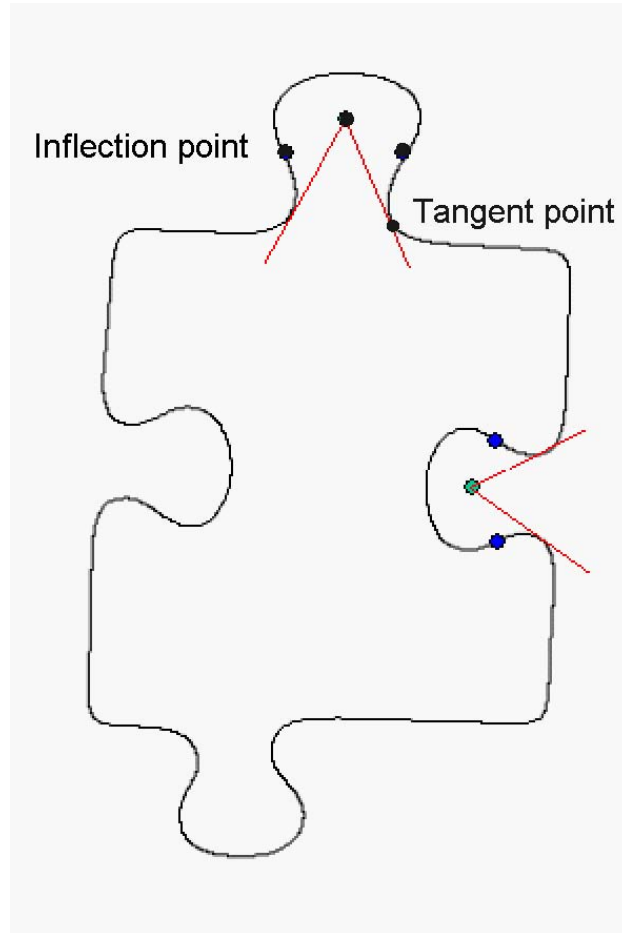


Figura 14: Punts seleccionats en una peça amb les dades parametritzades

L'inconvenient més significatiu a aquesta estructura de dades és que amb les transformacions que pateix la imatge des de l'adquisició fins a l'anàlisi, sovint no és suficient informació per a determinar si dues peces són contigües o no. Per aquest motiu es descarta aquesta opció.

2.3.2.1.2. Codificació dels costats

El fet d'emmagatzemar tota la informació del contorn en una cadena de caràcters té un inconvenient important, que és l'espai de memòria que ocupa. D'altra banda, dins dels avantatges comptem amb que es pot fragmentar fàcilment i permet la reconstrucció de la imatge original de manera directa. Un valor afegit a això és que existeixen diferents algorismes de comparació de cadenes.

La codificació escollida per a passar de la imatge del contorn a la cadena de caràcters és el *Chain Code de Freeman*, que s'explica amb més detall dins de **3 Sistema de Resolució** a la secció **3.2 Codificació i Comparació de la imatge**.

2.4. *Matching*

Un cop tenim les peces emmagatzemades en una estructura, el procés de *Matching* és el que posa en correspondència aquestes peces donant com a resultat final una solució que permeti la reconstrucció del puzzle. Determinar quin és l'algorisme que retorni aquest resultat amb el menor error possible i la major velocitat és pràcticament impossible.

2.4.1. Tècniques Satisfacció de Restriccions / AC-4

L'AC-4, *Arc Consistency 4*, és un algorisme molt utilitzat i considerat òptim dins del camp de la Intel·ligència Artificial. El seu funcionament consisteix a construir un camí entre conjunts d'elements que satisfaci una sèrie de restriccions. L'eficàcia d'aquest algorisme resideix en la rapidesa de propagar les restriccions i eliminar comparacions en un estat molt inicial.

La complexitat que comporta l'adaptació d'aquest algorisme per al projecte és un punt pel qual es descarta aquesta opció. També hi ha el fet que sovint els puzzles poden tenir diferents resolucions, si ens basem exclusivament en la seva forma, de manera que s'han d'aplicar altres tècniques de comparació un cop tenim una primera aproximació de solució.

2.4.2. Força Bruta

Com el seu nom indica, aquest mètode consisteix en comprovar la possibilitat de correspondència de tots els costats de cada peça amb tota la resta, acceptant aquella parella de costats que maximitzi el valor de semblança entre ells.

El mètode implementat està a mig camí entre una tècnica i l'altra. Inicialment només s'han contemplat els casos dels costats que són de tipus contrari, *Indent* amb *Outdent* i viceversa. Una altra consideració és que les peces *Contorn* tan sols s'aparellen entre elles pels costats contigus al costat *Recte*. D'aquesta manera, eliminem una sèrie de comparacions que sabem que no poden donar-se. Tot i això, dins dels casos possibles es fa una comparació de tots amb tots.

3. Sistema de Resolució

Partint del que s'ha exposat a 2, en aquesta secció s'entrarà a explicar amb més detall els passos que s'han seguit per a desenvolupar tot el sistema fins arribar a la conclusió. El procés que es segueix parteix de l'adquisició de les peces amb un escàner, continua amb la detecció i el retall de cada peça, s'analitzen i es construeixen les estructures de dades, es posa en correspondència cada peça i finalment es mostra per pantalla el resultat final obtingut.

La imatge 15 mostra els passos que es segueixen i el que se n'obté després de cada pas.

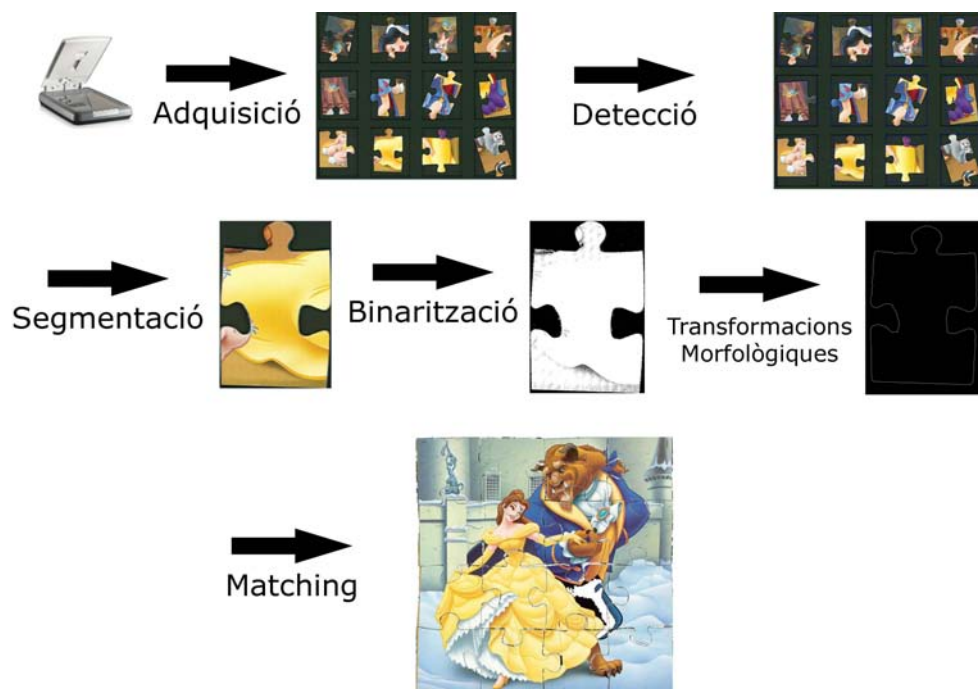


Figura 15: Fases del sistema de resolució

3.1. Adquisició i segmentació d'imatges

Aquest és el punt inicial de tot el projecte. Partim d'un puzzle físic i l'objectiu és aconseguir una imatge per a cada peça d'aquest.

3.1.1. Adquisició

Aquest pas és molt important, doncs és a partir del qual es construeix tot el projecte i d'on sortiran les imatges per a ser tractades i per a ser mostrades al resultat final. Per aquest motiu és important que tinguin els menors errors possibles.

El soroll més fàcilment controlable és el que provoca la pols. Per un costat, amb la neteja de la superfície de l'escàner i de les peces, se n'evita una gran part. Per un altre costat, amb el tractament via software es pot arreglar en gran mesura.

L'altre gran problema és l'inevitable de les ombres que tot escàner genera sobre un objecte que tingui un volum, per petit que sigui. Generalment la tapa de l'escàner genera el fons de la imatge blanc i té la conseqüència que l'ombra queda molt marcada. Per a suavitzar-ho es tracta de trobar un color que absorbeixi la llum i reflecteixi poc. Per definició, aquest color és el negre. Per tant, utilitzant un cartró o cartolina negra i posant-ho com a fons de la imatge, s'aconsegueix minimitzar l'ombra. La figura 16 mostra la comparació de l'aparició d'ombres en funció del color de fons.

Un tercer inconvenient de l'adquisició d'imatges en color és la possible confusió de part de l'objecte amb el fons. Però aquest problema no té fàcil solució. Després de realitzar algunes proves, s'opta per resoldre aquesta part durant el tractament de la imatge en el procés de segmentació.

3.1.2. Segmentació

Disposem d'una imatge amb un fons homogeni i un nombre indeterminat d'objectes. L'objectiu és generar una imatge per a cadascun d'aquests objectes i que sigui de tamany mínim.

3.1.2.1. Detecció de l'objecte

Una de les maneres de discriminar un objecte sobre un fons llis és l'ús de llindars o *thresholds*. Existeixen molts algorismes implementats per als llindars. Els millors resultats obtinguts dins les proves realitzades per al problema plantejat, han sigut els de l'algorisme de *Otsu*.

Aquest mètode parteix de la base que a la imatge hi ha dos tipus de punts, els del objecte i els del fons. Amb aquesta premisa, busca automàticament, a partir de l'histograma de color, el valor que minimitza la variança dintre dels dos grups de pixels.

Un cop tenim el valor del color que fa de llindar entre un grup i l'altre, l'apliquem sobre la imatge, obtenint-ne una de binaritzada, i utilitzant les

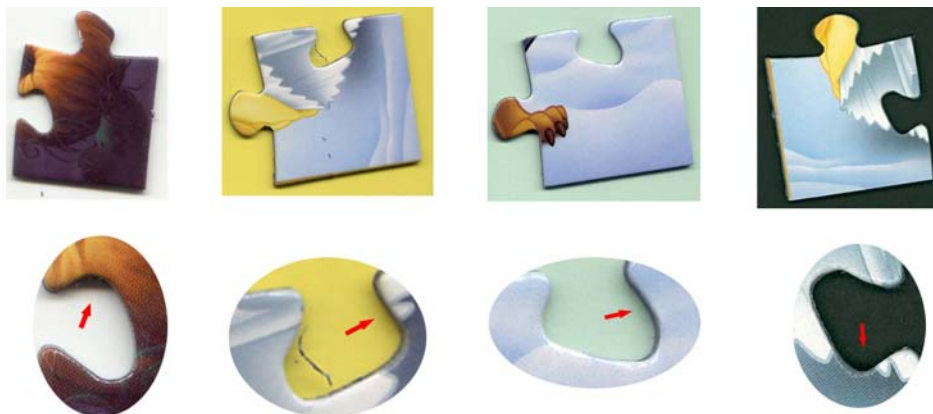


Figura 16: Ombres produïdes durant el procés d'escaneig

funcions d'OpenCV busquem el quadrangle mínim que engloba cada peça, *Bounding Box*. La figura 17 mostra el resultat d'aquest procés, aplicant el requadre obtingut sobre la imatge original.



Figura 17: Resultat de la detecció de peces

3.1.2.2. Retall de l'objecte

La importància de la separació entre peces a l'hora d'escanejar queda reflectida en aquest punt, ja que dues peces molt pròximes poden acabar considerant-se una, un cop aplicat el lllindar i durant la búsqueda del *Bounding Box*. Tan bon punt tenim la regió de la imatge que circumscriu cada peça, la definim com a Regió d'Interès, *ROI*, i la copiem en una imatge nova, obtenint-ne així una per a cada una de les peces.

3.2. Extracció de Característiques

Aquest procés dóna forma a tot el projecte, doncs és on es defineix quin tipus d'informació necessitem de la peça i es prepara tota l'estructura que servirà per al procés de comparació i *matching*. En aquest punt es decideix si es fa un anàlisi per color i/o per forma.

En el nostre cas tan sols fem l'anàlisi per forma, de manera que el primer que hem d'aconseguir és el contorn de la peça. Per tant hem de realitzar una sèrie de transformacions morfològiques per tal d'eliminar la resta de soroll que hagi pogut quedar després del *threshold*.

Primer passem la imatge de color a blanc i negre. Un cop binaritzada, amb una dilatació i una erosió inicial (*Closing*), eliminem part dels objectes externs a la peça. Tot seguit fem una búsqueda del contorn amb la funció `cvFindContour` de OpenCV. Aquesta funció ens retorna una sèrie de conjunts de punts que engloben els objectes de la imatge. Com el nostre objectiu és el contorn de la peça, que és el bloc més gran, només ens hem de quedar amb el grup que conté més punts. La figura 18 és el resultat d'aquest passos.

Per tal de facilitar els processos següents, es reorienten les peces de manera

que el quadrangle generat al voltant d'elles sigui el mínim possible. Per aconseguir això, es fa ús de la funció `cvMinAreaRect2` de la que disposa OpenCV, que ens retorna l'angle de rotació que s'ha d'aplicar a la imatge per a obtenir el resultat que busquem, tal i com mostra la figura 3.3.

El següent pas del procés és la detecció de cada costat de la peça per a poder-lo classificar segons la seva forma. Per a poder definir cada costat del

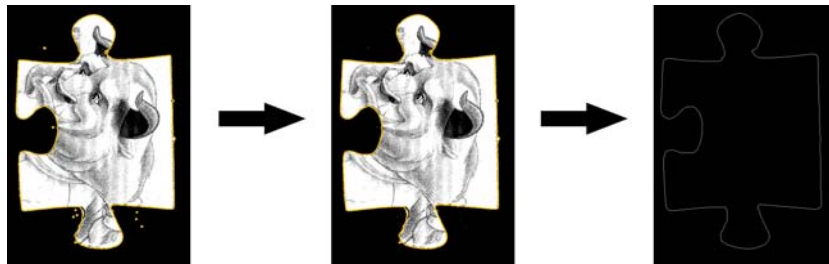


Figura 18: Passos per a obtenir el contorn de la peça

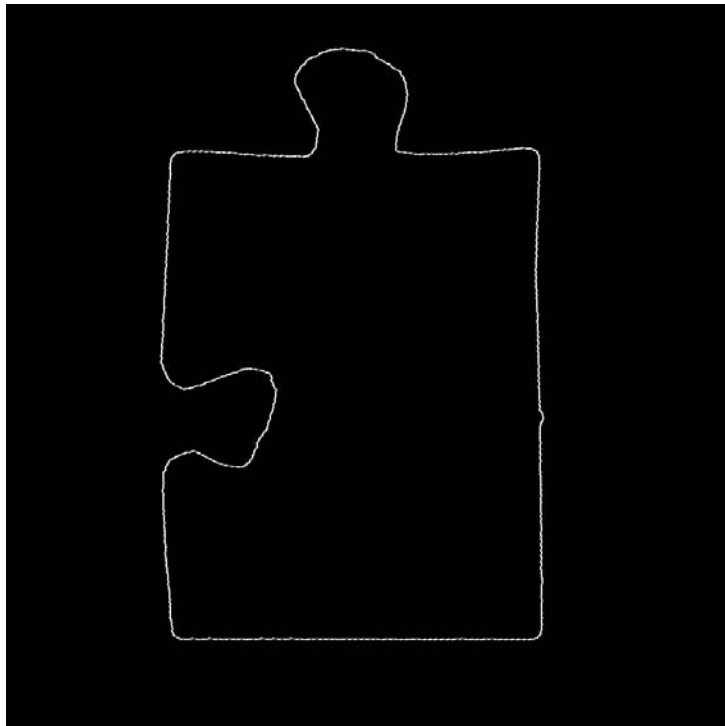


Figura 19: Contorn d'una peça retallat i orientat

contorn, el primer que s'ha de fer és buscar els quatre punts que fan de nexa entre dos costats. Sabent que tenim la imatge amb les peces correctament orientades, podem dividir la imatge en quatre parts iguals i podem assegurar que hi ha una cantonada per cada sub-imatge.

Podem utilitzar diferents fòrmules per a trobar aquest punt. La que hem fet servir en aquest cas és la proposada a l'article [3], fent un escombrat amb una recta diagonal, començant de l'extrem perpendicular a la recta, fins a trobar el primer punt de l'objecte. A la figura 20 es mostra aquest procés, marcant amb un punt les *cantonades* obtingudes i amb una fletxa vermella el recorregut de cada línia de l'escombrat.

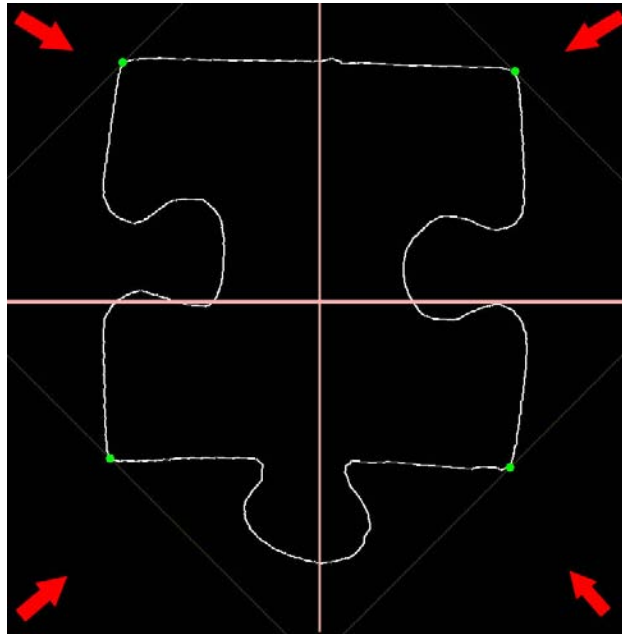


Figura 20: Resultat de la detecció de punts *cantonada*

Un cop tenim els quatre punts, cada un dels quatre costats és la successió de punts continguts entre cada parella de punts contigus. Finalment, hem de classificar aquests segments segons els tres possibles casos: *Indent*, *Outdent* o *Recta*.

Els *Indents* són aquells costats que tenen la major part dels punts desplaçats cap a l'interior de la peça respecte la línia recta que uneix els dos punts dels extrems del mateix costat. Els *Outdents* són els oposats als *Indents*, és a dir, els punts estan més allunyats de l'interior de la peça. I per últim les *Rectes* són aquells costats els punts dels quals no varien gaire respecte la recta traçada entre els punts extrems del segment.

El procés per a reconèixer i classificar cada costat és simple. Agafant una mostra prou significativa de cada segment i calculant la distància a una recta traçada entre els punts extrems, sabem en quin dels casos estem. Ja amb la classificació feta, trobem tres tipus de possibles peces: *Cantonades*, *Contorn* i *Centre*. Del primer cas tan sols n'hem de trobar quatre peces, i són aquelles que tenen dos costats contigus de tipus *Recta*. El segon cas, disposa d'un costat tipus *Recta*, i l'últim no en disposa de cap. Un exemple d'aquestes definicions es pot veure a la figura 21

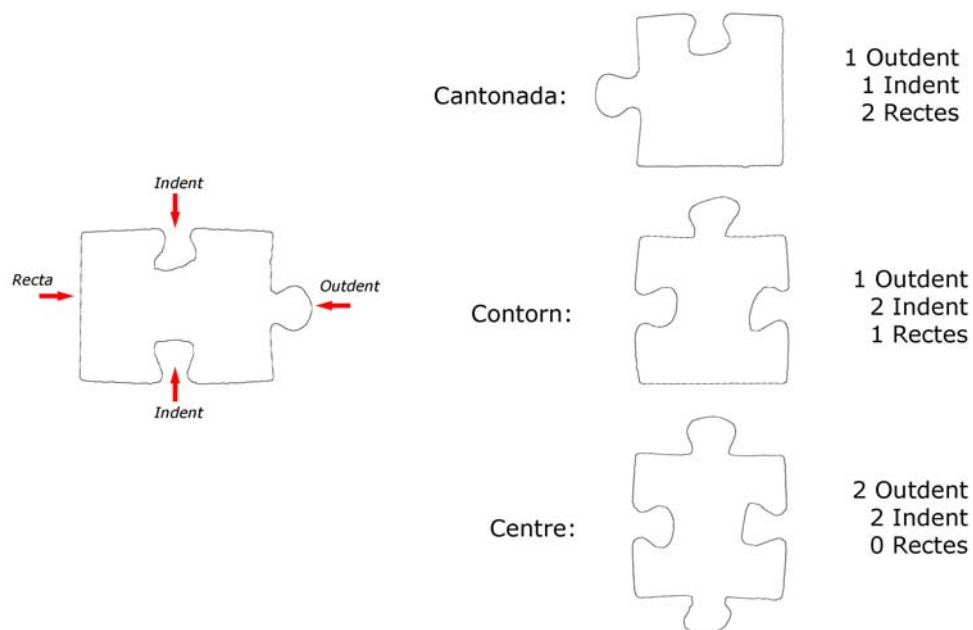


Figura 21: Mostra i definició utilitzada a les peces

3.3. Codificació i Comparació de la imatge

Arribats a aquesta fase, ja tenim totes les dades de la peça. El que necessitem ara és muntar l'estructura de dades que ens servirà per a poder comparar les peces entre sí i poder fer el procés de *Matching*.

3.3.1. Codificació

Previ a la comparació de les peces, s'ha de definir una codificació per tal de tenir el mateix sistema de valors entre peces. Dintre de les múltiples possibilitats, s'ha optat per la codificació *Chain Code de vuit punts* de *Freeman*, ja que permet una reconstrucció ràpida de la imatge, facilita les tasques de comparació i permet fer les operacions de rotació de manera ràpida i senzilla.

La codificació *Chain Code* parteix d'un punt i recorre la imatge en el sentit de les agulles del rellotge. Per a cada punt, busca dins les vuit coordenades on cau el següent punt. Per a cada coordenada s'ha assignat un valor que serà constant per a tota la imatge. De manera que el resultat final és una successió de números que indiquen la direcció del punt següent de l'objecte.

A la imatge 22 es mostra com es codifica amb aquest mètode. A la part esquerra es mostra els valors utilitzats a cada direcció i la part dreta una simulació del resultat de la codificació de la imatge.

3.3.2. Estructura de Dades

La manera com s'ha emmagatzemat la informació que defineix la peça és important per a poder fer el procés de comparació i per a poder reconstruir la solució al final.

Per a la resolució farem servir tres estructures: la *Peça*, el *Mapa de peces*

i el *Mapa de costats*.

3.3.2.1. Peça

Inicialment tenim la ruta per a localitzar la imatge en color guardada a disc, que ens servirà per a poder mostrar el resultat final.

Tot seguit guardem l'angle de rotació que s'ha d'aplicar a la imatge per tal de tenir-la ben orientada. Aquest valor s'ha obtingut durant la Extracció de Característiques.

Partint del contorn codificat amb el *Chain Code* i els quatre punts cantonades, emmagatzemem quatre vectors, corresponent a cadascun dels costats codificats. Cal tenir en compte l'ordre en què s'emmagatzema, doncs cada costat ha de ser contigu amb l'anterior i el següent. A més a més, per mantenir una lògica, el que es considera el primer dels costats és el que està a la part més superior de la imatge i es segueix en sentit antihorari per la resta.

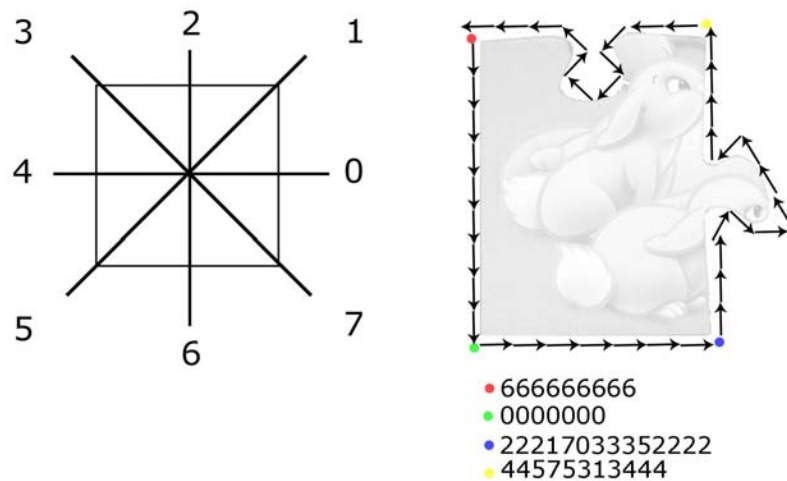


Figura 22: (Esquerra) Valors assignats a cada orientació. (Dreta) Resultat de la codificació de la imatge

Finalment, per a cada costat es disposa d'una etiqueta que identifica si aquell costat és un *Indent*, un *Outdent* o una *Recta*. I en funció de la quantitat de *Rectes* de les que disposi la Peça, un altre etiqueta defineix el tipus de Peça, podent ser *Cantonada*, *Contorn* o *Centre*.

3.3.2.2. Mapa de Peces

Durant el procés de comparació, es construeix una matriu de $M \times N$. Per a cada fila i columna s'hi va posant el valor identificador d'una estructura Peça de manera que es pugui saber fàcilment quines estan en correspondència amb quines.

3.3.2.3. Mapa de costats

Per tal de facilitar el procés tant de comparació com de reconstrucció del resultat final, es construeix una segona matriu de les mateixes dimensions que el Mapa de Peces, on es guarda l'índex del costat que ha d'estar a la part inferior per estar en correspondència amb les peces del voltant. Això ens ajuda a saber quantes rotacions de 90 graus s'han de donar a la peça per al muntatge final.

3.3.3. Comparació

Amb les dades de les peces emmagatzemades, hem de fer el procés de comparació entre elles per a saber quines són les que tenen major grau de similitud, és a dir, les màximes candidates a estar en correspondència.

Inicialment començarem per una peça etiquetada com a *Cantonada*. Si disposem de peces *Centre* podem afirmar que dues *Cantonades* no estaran

junes. Aquesta premisa ens serveix per a poder descartar comparacions entre *Cantonades*.

Seguidament, i per a facilitar el procés, agafem totes les peces *Contorn* i busquem el seu costat *Recte*. Seguint el sentit antihorari, el costat següent al trobat el considerarem el costat *dret*, i l'anterior, l'*esquerre*. Aquesta mateixa consideració també és la que es fa servir per les peces *Cantonada*, tenint en compte que hi ha dos costats *Rectes* seguits.

Per aconseguir el marc més extern del puzzle, hem de comparar els costats *drets* de les peces amb els *esquerres* i trobar aquell que té major grau de semblança. Per descartar casos, tan sols es compararan els *indents* amb els *outdents*, doncs físicament no es poden encaixar dos costats del mateix tipus.

El càlcul del grau de similitud s'ha desenvolupat fent una variació de l'algorisme de la *Distància de Levenshtein*. Donades dues cadenes, l'algorisme associa un cost a cada una de les possibles operacions que s'ha de fer a una d'elles per igualar-la a l'altra. Aquestes operacions són tres: insertar un nou caràcter, eliminar-lo o modificar-lo.

L'adaptació que s'ha fet pel cas concret dels puzzles té en compte diferents aspectes. Partim de la base que el que estem comparant són cadenes de valors compresos entre el zero i el set que responen al *Chain Code* del costat, explicat al punt anterior. Una de les consideracions que s'ha tingut és que, ja sigui per alteracions de la imatge en tots els processos previs o perquè el puzzle s'ha creat així, les longituds dels costats no tenen perquè ser iguals. Així que, tant la inserció com l'eliminació d'un valor tindrà un cost igual en ambdós casos però diferent a la modificació. En aquesta última acció és on s'ha fet la major modificació de l'algorisme. Donat que el que comparem són direccions, no és el mateix canviar un valor per un altre que segueix prenent la mateixa

Accio		Cost
Insertar		0.5
Eliminar		0.5
Modificar	± 1	0.5
	± 2	1
	± 3	1
	± 4	1

Taula 1: Costos assignats per a cada transformació dins l'algorisme de Levenshtein

direcció que per un d'oposat. Així que s'ha assignat costos diferents, segons el grau de diferència entre un valor i l'altre de la cadena. La taula 1 mostra els valors utilitzats.

Aquest algorisme ens torna un valor entre zero i u, indicant la mesura de diferència entre costats. A partir d'aquí es poden detectar les peces candidates a ser contigües i descartar aquelles amb el valor més elevat.

Pel cas de les peces *Centre*, el procés de comparació és el mateix, tenint en compte que a la que es disposa de la part més externa del puzzle, trobem extrems on les peces a encaixar han de complir dos requisits de costats. I per cada nova peça que es posa, apareixen dos nous extrems, com s'exemplifica a la figura 23.

Això serà així fins arribar a la última fila i columna, on les peces a comparar hauran de complir els requisits de tres costats.

3.4. *Matching*

El procés de *Matching* és la resposta a la comparació de les peces. A partir dels valors anteriors, es va construir el Mapa de Peces i el de Costats, prioritzant aquelles peces que tenen un valor de semblança superior.

Però tal i com s'explica a l'anàlisi de requeriments, dues peces diferents poden tenir la mateixa forma i conseqüentment estar en correspondència amb múltiples peces. En el nostre cas implica que no sempre el valor menor obtingut durant la comparació sigui el contigu en aquell costat.

Per això en aquesta part es construeixen diferents Mapes de Peces que són solució al puzzle i, a partir dels valors de la comparació entre peces, es dona aquella que té una diferència global menor, segons l'algorisme de *Levenshtein*.

3.5. Representació final

Un cop acabat el procés de comparació i correspondència de peces, ja tenim les dues taules que ens permetran construir la imatge final. L'últim

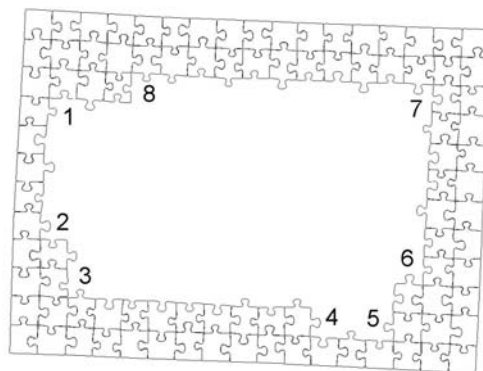


Figura 23: Il·lustració de com queda muntat el contorn i els extrems nous que apareixen

procés és recuperar la imatge inicial, rotar-la per tal que quedi ben orientada tenint en compte l'angle inicial i el Mapa de costats, ubicar-la en funció del Mapa de Peces i finalment mostrar-ho per pantalla com es veu a la figura 24.

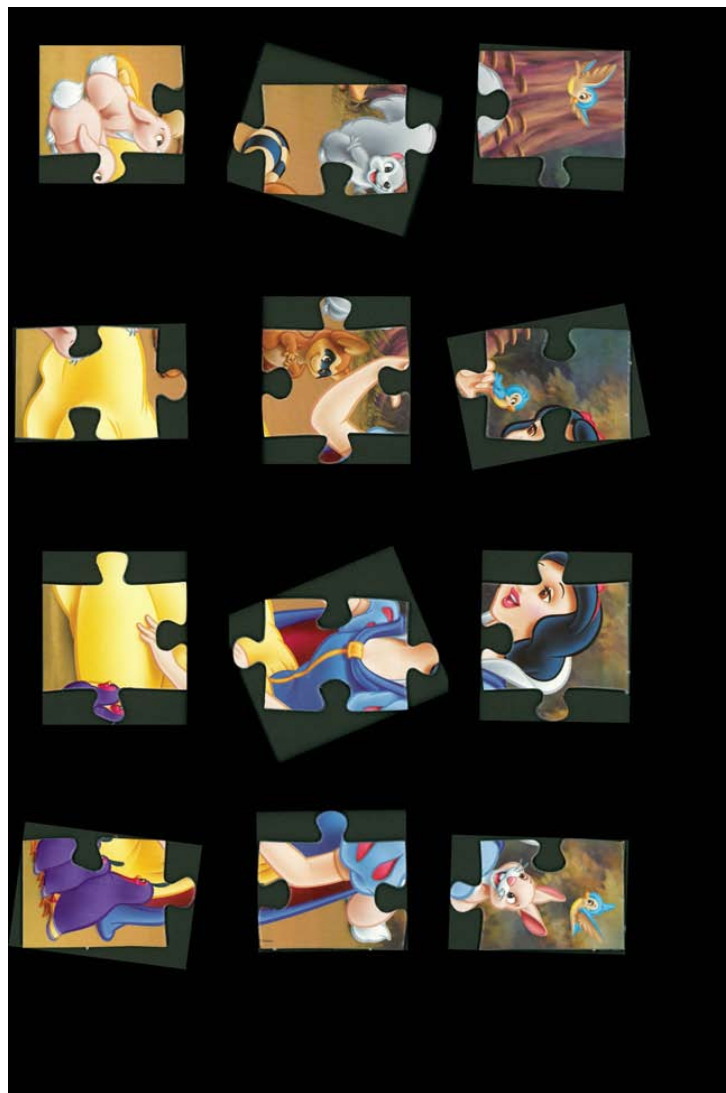


Figura 24: Resultat final de la resolució del puzzle

4. Resultats obtinguts

Les proves s’han realitzat en un ordinador portàtil, Intel Celeron a 2Ghz amb 1Gb de memòria RAM, amb Windows XP com a sistema operatiu. Els puzzles amb els que s’ha testejat el projecte van des de les dotze peces fins a les vint-i-cinc, amb il·lustracions de personatges Disney.

La següent taula mostra la resposta del sistema en cadascun dels puzzles.

Títol	Peces	Temps	Error Promig	Correcte
Blancaneus	12	17.171s	5.50 %	Sí
La Bella i la Bèstia	16	79.687s	4.725 %	Si
La Sireneta	20	2258.89s	11.38 %	No
Ventafocs	25	—	—	No

Títol és el nom que se li ha assignat al puzzle per a identificar-lo. Peces indica el nombre de peces que componen el puzzle. Temps està representat en segons i és l’estona que triga el procés de matching, partint de les dades correctament adquirides, per a donar un resultat que considera òptim. Sota la columna *Error Promig* tenim el tant per cent d’error promig que hi ha entre cada costat posat en correspondència. Aquest valor s’obté fent la mitjana dels resultats obtinguts amb l’algorisme de *Levenshtein* amb els *matchings* tant verticals com horitzontals. La última columna és el factor humà que revisa si el resultat mostrat és l’esperat o no.

Les imatges 25 i 26 mostren els dos puzzles solucionats correctament. En canvi, la figura 27 és el resultat erroni de *La Sireneta* i a la 28 es pot veure com no s’han detectat correctament les peces i per tant no s’ha pogut seguir amb la resta de processos.

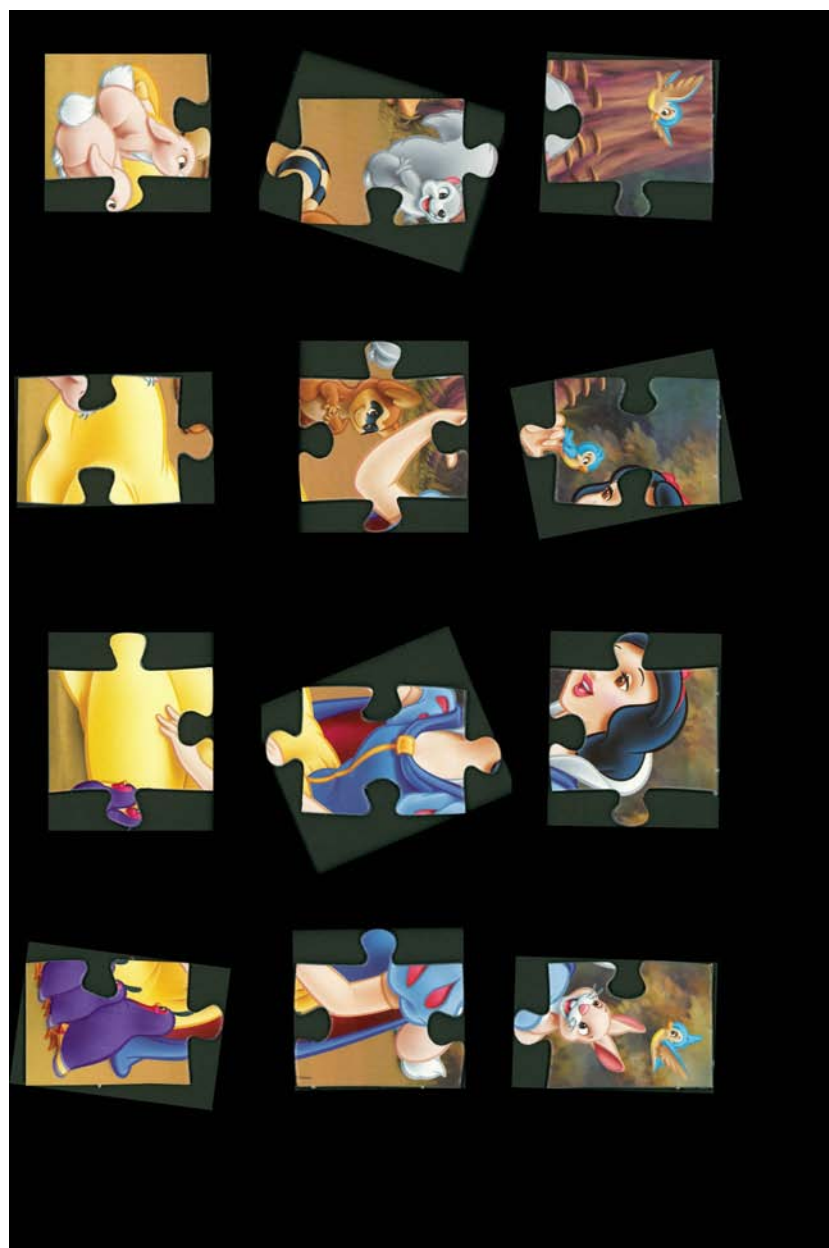


Figura 25: Puzzle Blancaneus resolt

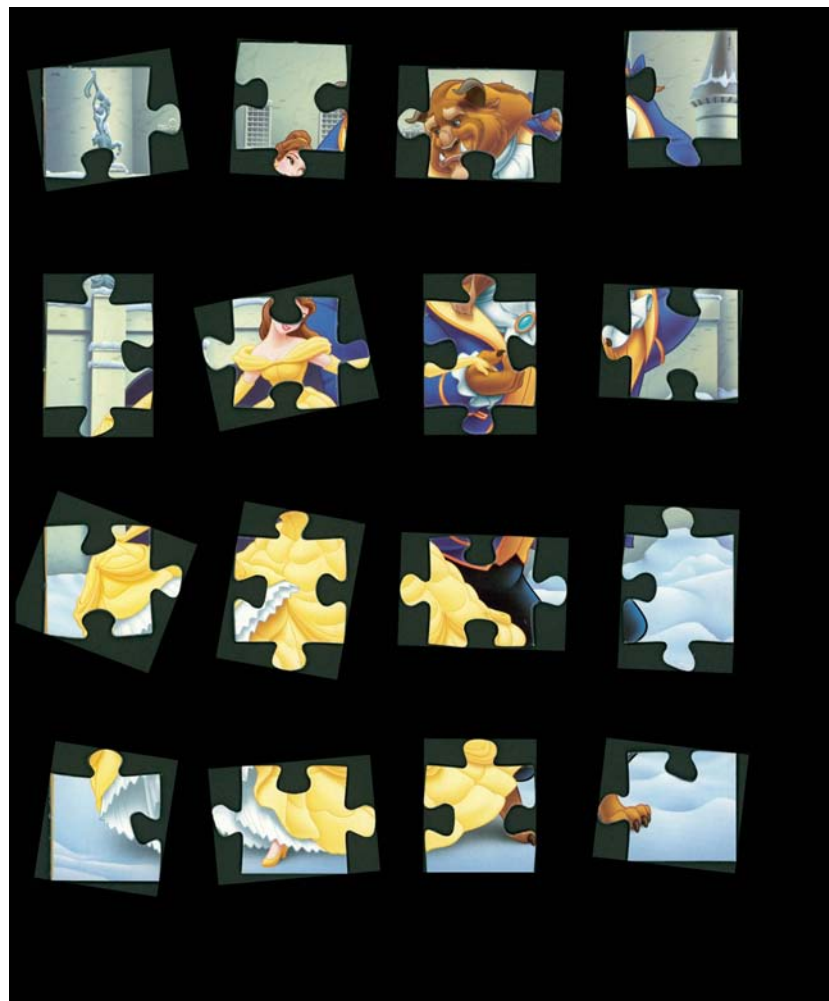


Figura 26: Puzzle La Bella i la Bèstia resolt

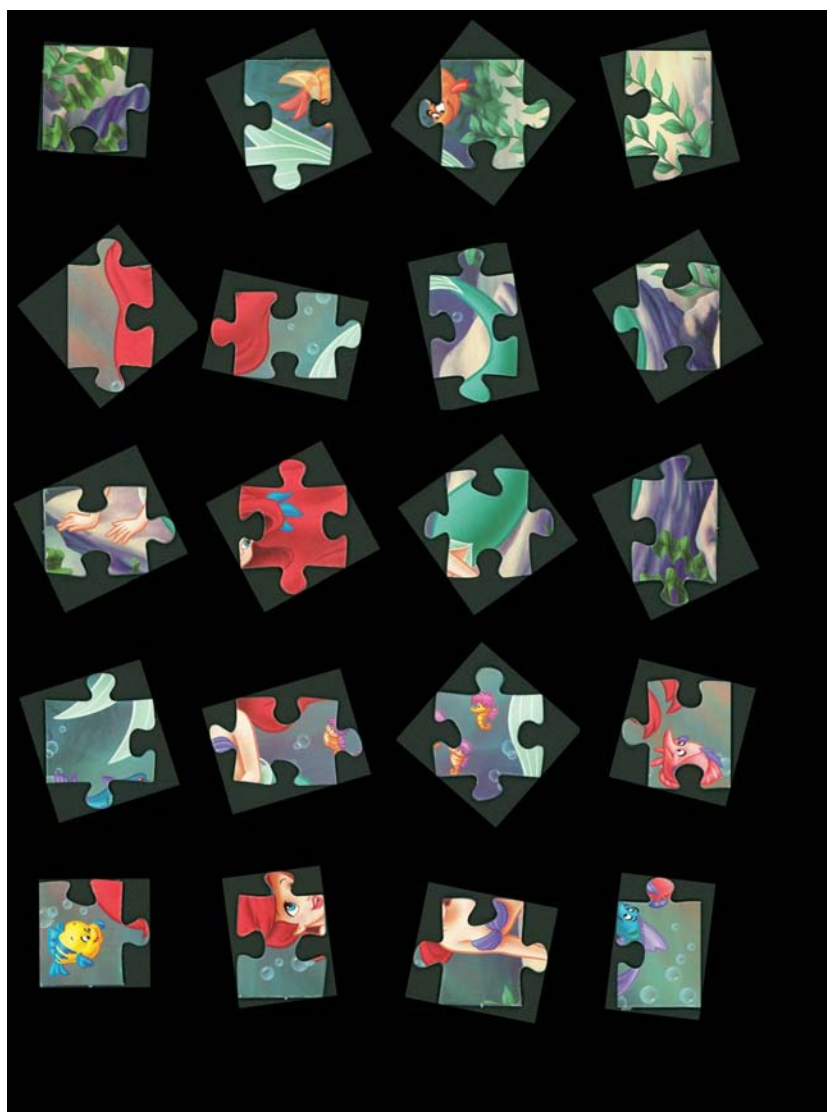


Figura 27: Puzzle La Sireneta mal resolt



Figura 28: Puzzle Ventafores amb mala detecció de peces

5. Conclusions i treball futur

El projecte presentat i exposat aquí és una possible solució que tan sols posa de manifest la gran quantitat de factors que intervenen en la recerca d'objectes aparellables, el cost computacional que impliquen i el temps que suposa. Si bé no és una tasca impossible, tampoc és un simple entreteniment muntar un puzzle de vint peces amb tècniques de visió per computador.

Com a treball futur i possibles ampliacions del projecte realitzat, el primer que podem implementar és la part del color a l'anàlisi de les imatges, ja que a part d'ajudar a discriminar casos, el reconeixement de traces i elements comuns entre peces podria facilitar les tasques de *matching*. En un segon pas, aquest anàlisi per color, pot fer que les pre-condicions de forma no siguin tan restrictives amb puzzles de *Indents*, *Outdents* i *Rectes*, i poder acceptar altres formes o fins i tot mosaics on les peces són quadrades i generalment de la mateixa mida.

Si a més de plantejar millores en el sistema d'anàlisi, pensem en ampliacions a nivell de codificació i comparació, el que es pot plantejar és la reconstrucció de qualsevol objecte independent de mida, forma i quantitat de peces, podent posar de manifest la falta d'algun tros i generant-lo virtualment. Per això, la comparació ja no seria entre segments (costats) sinò entre seqüències cícliques que poden posar-se en correspondència en qualsevol punt del contorn. Aquest podria ser el cas de documents fets malbé o deteriorats pel pas del temps, com per exemple, un paper arqueològic com el que es mostra a la imatge 29.



Figura 29: Fragment del papir Artemidoro

6. Bibliografia

- [1] M. G. Chung, M. Fleck and D. A. Forsyth “Jigsaw Puzzle solver using shape and color”
Proc. ICSP '98, pp. 877 - 880, 1998
- [2] D. Goldberg, C. Malon and M. Bern “A global approach to automatic solution of Jigsaw Puzzles”
SoCG2002, Vol. 28, pp. 165 - 174, 2004
- [3] Ture R. Nielsen, Peter Drewsen, Klaus Hansen “Solving jigsaw puzzles using image features”
Pattern Recognition Letters 29, pp. 1924 - 1933, 2008

RESUM

Aquest projecte és un anàlisi d'algunes tècniques de reconeixement d'objectes en 2D, la seva comparació i equivalència. La solució presentada en aquesta memòria ha estat implementada en C++ per a poder avaluar els resultats així com les diferents problemàtiques aparegudes durant tot el procés. La resolució de puzzles és tan sols un exemple dels múltiples casos d'objectes bidimensionals que ens podem trobar que necessiten tècniques de *matching* per obtenir un objecte global.

RESUMEN

Este proyecto es un análisis de algunas técnicas de reconocimiento de objetos en 2D, su comparación y equivalencia. La solución presentada en esta memoria ha sido implementada en C++ para poder evaluar los resultados así como los diferentes problemas que aparecen durante todo el proceso. La resolución de puzzles tan solo es un ejemplo de los múltiples casos de objetos bidimensionales que nos podemos encontrar que necesitan técnicas de *matching* para obtener un objeto global.

SUMMARY

This project is an analysis of some techniques of 2D object recognition, its comparison and equivalence. The solution presented in this report has been implemented on C++ to evaluate the results and the several problems that appear throughout the process. Solving jigsaw puzzles is just one example of the many cases of two-dimensional objects that we can find which need *matching* techniques to obtain a global object.